FTV360: a Multiview 360°Video Dataset with Calibration Parameters

Thomas Maugey, Laurent Guillo, Cédric Le Cam

FTV360: a Multiview 360° Video Dataset with Calibration Parameters

Thomas Maugey
Inria Rennes Bretagne Atlantique
Rennes, France
thomas.maugey@inria.fr

Laurent Guillo
CNRS, IRISA
Rennes, France
laurent.guillo@irisa.fr

Cedric Le Cam
Inria Rennes Bretagne Atlantique
Rennes, France

ABSTRACT

In this paper, we present a new dataset in order to serve as a support for researches in Free Viewpoint Television (FTV) and 6 degrees-of-freedom (6DoF) immersive communication. This dataset relies on a novel acquisition procedure consisting in a synchronized capture of a scene by 40 omnidirectional cameras. We have also developed a calibration solution that estimates the position and orientation of each camera with respect to a same reference. This solution relies on a regular calibration of each individual camera, and a graph-based synchronization of all these parameters. These videos and the calibration solution are made publicly available.

CCS CONCEPTS

• **Information systems → Multimedia databases**; • **Computing methodologies → Camera calibration**.

KEYWORDS

Datasets, 6DoF, Free Viewpoint Television, 360°, Calibration

ACM Reference Format:
Thomas Maugey, Laurent Guillo, and Cedric Le Cam. 2019. FTV360: a Multiview 360° Video Dataset with Calibration Parameters. In *10th ACM Multimedia Systems Conference (MMSys '19), June 18–21, 2019, Amherst, MA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3304109.3325815

1 INTRODUCTION

Free Viewpoint Television (FTV) is an emerging application in which a multimedia content is transmitted to users such as they are enabled to choose and change the viewing angle in real time [11]. In other words, users have the possibility to observe the scene from the viewpoint they want: either by choosing between a predefined set of views (Fig. 1(a)), or by freely navigating in the scene (Fig. 1(b)) and thus experiencing 6 degrees of freedom (6DoF).
The target applications are numerous: from the transmission of sportive and cultural events for television, to more immersive communication with augmented reality applications as the "industry 4.0", education and health.

Ultimately, FTV with 6DoF is a challenging scenario that faces multiple research issues, dealing with the whole processing chain.

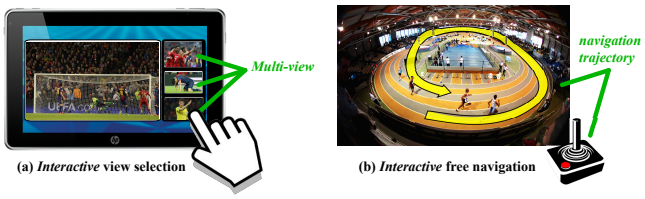

Figure 1: Two versions of FTV: (a) view switching, (b) free navigation

The data *representation* task consists in combining the signal captured by multiple heterogenous devices (*e.g.,* light field, 360, stereo cameras) and finding a proper description well adapted to the transmission constraints. Different compact representations have been investigated such as LDI [6, 10], GBR [7], mesh [1], point-clouds [3]. Then a dedicated *compression* scheme has to be designed. Contrary to "traditional" compression in which all the data is sent to the user, the specificity of FTV is that the transmission system has to transmit only what is required by the user. Indeed, it is not necessary to transmit all the scene when the user is only looking at one specific subpart of it. However, since online encoding is not conceivable, the main problem is to design a compression strategy that encodes all the data a priori such that only a subset could be extracted after the user request [4, 9]. At the user's side, efficient *rendering* algorithms are needed to make the navigation as smooth as possible, by artificially increasing the number of views.

Although developing competitive *representation, compression and rendering* solutions is already highly challenging, making the user experiencing a real 6 degrees of freedom (6DoF) navigation in the scene is nowadays impossible mostly because no proper *acquisition* system exists. Indeed, the *acquisition* for FTV presents several important issues. In order to enable a high quality navigation, it is required that the user has access to a number of viewpoints sufficiently large to cover a complete wide 3D scene. Even though some of them might be virtually synthesized, the initial capture must be sufficiently dense to enable good synthesis quality. This makes the acquisition process very costly in terms of hardware. Another issue is that the calibration algorithms used for small captured systems have to be revisited since the cameras may be more distant in a FTV context. Nowadays, datasets are either totally synthetic or only enable 3DoF navigation (*i.e.*, only the rotation of the head and a small 3D sensation).

In this paper, we propose a novel dataset made of 8 sequences (each of them contains 40 synchronized and calibrated 360° videos). This enables a 3DoF navigation *at many different places* in the scene, which comes down to a "discretized 6DoF" solution. We also

MMSys '19, June 18–21, 2019, Amherst, MA, USA

ACM ISBN 978-1-4503-6297-9/19/06…$15.00
https://doi.org/10.1145/3304109.3325815

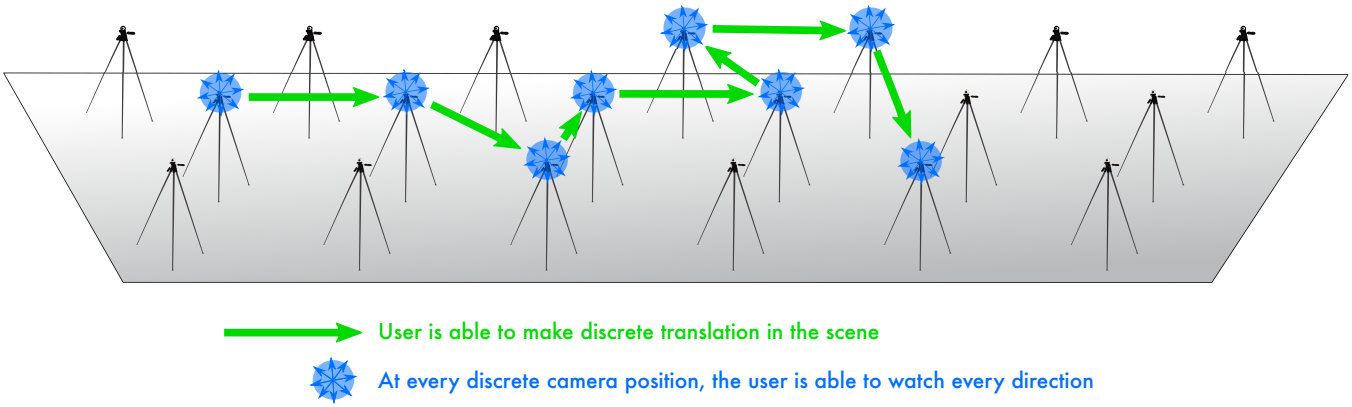

Figure 2: Proposed acquisition architecture for Free Viewpoint Acquisition.

present how this acquisition system is calibrated. First, we show that calibrating each camera with the unified spherical model [2, 8] is meaningful. Based on a parallel recording of a pattern moving in the scene, we explain how each camera can be calibrated with respect to this pattern. Finally, we propose a method that deduces from these multiple relative calibration parameters, a global position and orientation for each of the cameras. This novel dataset has completely new characteristics (high number of views, multiple 360°, wide and heterogeneous scene) that may greatly support researches on FTV or 6DoF user immersion.

In Sec. 2, we describe the proposed acquisition system based on 360^o video captures from multiple viewpoints. Then, in Sec 3, we describe our calibration framework. We finally detail our dataset in Sec. 4.

2 ACQUISITION PROCEDURE

Ultimate Free Viewpoint Navigation enables a user to freely change the position, $\mathbf{t} = [x, y, z] \in \mathbb{R}^3$, and the angle, $\mathbf{r} = [\alpha, \beta, \gamma] \in [-\pi/2, \pi/2] \times [-\pi, \pi] \times [-\pi, \pi]$ of his viewpoint. Naturally, it is impossible in practice to sample at every position the light rays coming from every direction. The challenge for an acquisition system is yet to make this sampling as dense as possible. For that purpose, perspective cameras have shown their limitation since each of them captures the light rays at one given position coming from one fixed subset of directions. Recently, omnidirectional (or 360°) cameras have been introduced in the public market. Their strength is that they are able to record the light rays at one given position coming from every direction, at a price of a possibly decreased angular resolution since the field of view is increased.

This has motivated the following proposed acquisition procedure : 40 omnidirectional cameras have been spread inside a scene and synchronously film its content. If a user is navigating through the recorded video, he/she is able to discretely translates in the scene, *i.e.*, $\mathbf{t} \in \{\delta_i\}$, and at each translation position, he/she is able to visualize the angle he/she desires, i.e., $\forall\ i,\ \mathbf{r}(\delta_i) \in [-\pi/2, \pi/2] \times [-\pi, \pi] \times [-\pi, \pi]$ (see Fig. 2). Such acquisition system has never been studied and seems promising since it gets closer to full 6DoF navigation. As such, it is considered to constitute an interesting starting point for further research.

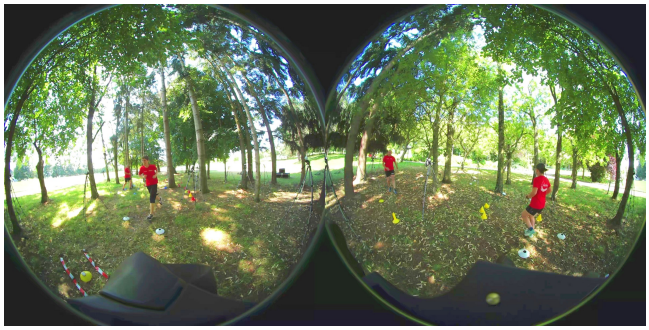

Figure 3: The used 360^o cameras capture two hemispherical images leading to an image at a resolution of 3840×1920.

In order to keep the data representation as close as possible to what is recorded by the camera (for calibration purposes), we keep the raw data format. In our solution, the videos are shot with Samsung Gear 360 cameras. They are made of two fisheye lenses spanning a bit more than 180° and placed at a distance of 4.5 cm. The raw footage consists of the image captured by the two lenses, without any geometrical correction, written side by side on the same frame. The resolution of the two hemispherical images put side by side is 3840×1920 at 30 fps, as shown by the example in Fig. 3. All the videos are synchronized manually at the frame precision. The calibration procedure that was used to estimate the relative position of each camera is explained in the next section.

3 CALIBRATION

3.1 Internal parameters estimation

The specificity of omnidirectional cameras is that their projection rules strongly depend on their architecture. As an exemple, catadioptric cameras (using a curved mirror to make the vision omnidirectional) do not project a light ray on its sensor as a fisheye lens does. In [2, 8], a general projection model has been proposed: the so-called *unified Spherical Model*. It is illustrated in Fig. 4. Let $\mathbf{P} = [X, Y, Z]^\top$ be a point in the 3D world expressed with respect to a world coordinate system centered in O. The unified spherical

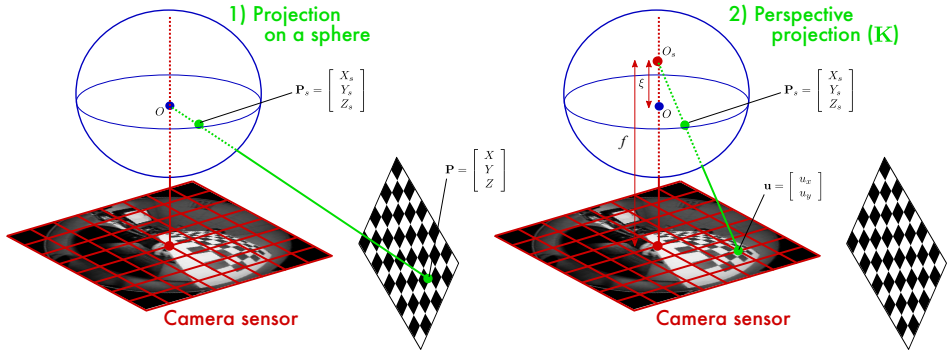

Figure 4: Unified spherical model adopted for the calibration of the camera used during our acquisition.

model makes a first projection of $\mathbf{P}$ on a sphere of radius 1 centered in O. The projected point $\mathbf{P}_s = [X_s, Y_s, Z_s]^\top$ can be deduced from $\mathbf{P} = [X, Y, Z]^\top$ by the following relationship :

$$\mathbf{P}_s = \frac{\mathbf{P}}{||\mathbf{P}||}. \tag{1}$$

Let us now define a point O_s that is the translation of O along the Z axis. The distance between O and its translation O_s is denoted by ξ. The unified spherical model then performs a second projection from the sphere to a plane that is parallel to the X and Y axis, and placed at a distance f from O_s. This second projection is a perspective projection, as in the traditional pinhole cameras. The projection $\mathbf{p} = [x, y]$ of point $\mathbf{P}_s$ onto the sensor array reads :

$$\begin{cases} x = f\frac{X_s}{Z_s+\xi} \\ y = f\frac{Y_s}{Z_s+\xi} \end{cases} \tag{2}$$

As a result, the pixel coordinates in the image plane can be written :

$$\begin{cases} u_x = k_x \left(\frac{fX}{Z+\xi\sqrt{(X^2+Y^2+Z^2)}} \right) + k_x x_0 \\ u_y = k_y \left(\frac{fY}{Z+\xi\sqrt{(X^2+Y^2+Z^2)}} \right) + k_y y_0 \end{cases}. \tag{3}$$

As mentioned in [8], the unified spherical model is sometimes completed with a radial and tangent distortions terms parametrized with 5 parameters $\{k_1, \ldots, k_5\}$. Here, we neglect these parameters (*i.e.,* $k_1 = \ldots = k_5 = 0$) since they are not necessary to model the projection properly in our system. A calibration algorithm aims at estimating the projection parameters k_x, k_y, f, ξ, x_0, y_0. For this paper, we have used the algorithm proposed in [8] and that is implemented in the OpenCV library. In Table 1, we show that the unified spherical model gives good results for our fisheye cameras. We also show that the same parameters can be used for every lens of all cameras. We precise that, in the following, the two lenses are treated separately, and thus considered as two different cameras (each one have its own set of parameters).

3.2 Individual external parameters estimation

In addition to the intrinsic values, the output of the calibration algorithm mentioned in the previous section contains the extrinsic parameters: a rotation and a position vector, respectively denoted by $\mathbf{t}$ and $\mathbf{r}$. They describe the position of the camera with respect to the pattern that is recorded. We thus record long calibration sequences where a pattern is synchronously filmed by all cameras. The pattern used in our experiments is a chessboard. To run the calibration of each camera we select the frames in which the pattern is visible. The output of such joint calibrations are:

- A time instant set $\mathcal{I}$ giving the frame timestamps during which the pattern is visible in at least one camera.
- For each, camera i, a time instant set $\mathcal{I}_i \subset \mathcal{I}$ corresponding to the time instants during which the pattern is visible in camera i.
- A set of positions $\{\mathbf{t}_n^i\}$ and rotations $\{\mathbf{r}_n^i\}$, where $n \in [\![1, |\mathcal{I}|]\!]$ and each $\mathbf{t}_n^i \in \mathbb{R}^3$ and each $\mathbf{t}_n^i \in \mathbb{R}^3$ are sorted in the time instant order.

The rotation angles $\mathbf{r}$ are given in the Rodrigues format [5]. The vector $\mathbf{r}$ gives the rotation axis, and $||\mathbf{r}||_2$ is rotation angle. The remaining step consists in gathering all these asynchronous relative positioning in order to deduce the position of all cameras in a fixed world coordinate system. For that aim, we propose an algorithm that is described in the next section.

	Cam. 1 front	Cam. 1 rear	Cam. 2 front	Cam. 2 rear
(cam)	0,85	1,13	1,37	1,45
(av.)	1,11	1,36	1,50	1,57

Table 1: Example of MSE computed between the pixel position measured and the ones estimated thanks to the unified spherical model parameters, for a camera specific parameters (cam) and averaged camera parameters (av.). This has been tested on all our cameras.

3.3 Global external parameters estimation

The challenge of this part is to gather all the parameters, relative to the moving chessboard pattern, and to deduce camera parameters. We first compute a correspondance matrix counting the co-occurence of cameras in the pattern detection. Concretely, for two cameras i and j, the matrix value is equal to:

$$\mathbf{G}(i, j) = |\mathcal{I}_i \cap \mathcal{I}_j|. \tag{4}$$

The matrix element $\mathbf{G}(i, j)$ indicates how much the joint calibration of two cameras i and j is reliable. Indeed, the calibration gives better results if more points are used in the optimization. We then choose the reference camera i_{ref}, from which all the other camera positions

will be set. We pick the most reliable one:

$$i_{\text{ref}} = \text{argmax}_i \sum_j \mathbf{G}(i,j). \quad (5)$$

The camera i_{ref} is thus the camera having the highest number of simultaneous calibration parameters with the other cameras.

The next step consists in setting: $\mathbf{t}^{i_{\text{ref}}} = \mathbf{0}$ and $\mathbf{r}^{i_{\text{ref}}} = \mathbf{0}$. Let us now denote by $\mathcal{N}(\mathbf{t}_{i_{\text{ref}}})$ the neigborhood of camera i_{ref}, *i.e.* the set of cameras that have joint calibration parameters. We denote a coordinate change as:

$$\begin{aligned} \phi_{i\to j}: \quad & \mathbb{R}^3\times\mathbb{R}^3\times\mathbb{R}^3\times\mathbb{R}^3 \quad \to \mathbb{R}^3\times\mathbb{R}^3 \\ & \mathbf{r}^i, \mathbf{t}^i, \mathbf{r}^j, \mathbf{t}^j \quad\quad\quad\quad \to \mathbf{r}^{i|j}, \mathbf{t}^{i|j} \end{aligned} \quad (6)$$

where $\mathbf{r}^{i|j}$ and $\mathbf{t}^{i|j}$ are the coordinates of camera i expressed in j coordinate system. We now do the following operation

$$\begin{aligned} &\forall i \in \mathcal{N}(\mathbf{t}_{i_{\text{ref}}}),\ (\mathbf{r}^{i|i_{\text{ref}}}, \mathbf{t}^{i|i_{\text{ref}}}) = \\ &\textstyle\sum_{n\in\mathcal{I}_i\cap\mathcal{I}_{i_{\text{ref}}}} \phi_{i\to i_{\text{ref}}}(\mathbf{r}_n^i, \mathbf{t}_n^i, \mathbf{r}_n^{i_{\text{ref}}}, \mathbf{t}_n^{i_{\text{ref}}}) \end{aligned} \quad (7)$$

In other words, we average all the joint estimated positions converted to the i_{ref} coordinate system. Then the algorithm repeats this for the most reliable camera in this neighborhood. It seeks its own neighborhood and estimate this position, converted to the i_{ref} coordinate system. These operations are repeated until one has an estimate of all the positions with respect to a single camera i_{ref}.

3.4 Calibration validation

In this section, we show that the proposed calibration strategy gives meaningful distance and orientation estimation for the cameras. In a first quantitative test, we position two cameras at different distances (measured with a laser), and we compare the results of the calibration algorithm. In Table 2, we show the distance evaluation remains close to the true one, even when the baseline between cameras is large (5 m).

Δx real (m)	1	2	3	4	5
Stand. Dev. of estimated Δx (m)	0,02	0,03	0,07	0,03	0,09

Table 2: Standard deviation between Δx estimated as a function of the true Δx between the two cameras.

Then, we present qualitative results in which, we show the estimated cameras positions, when 40 of them are positioned in a scene. The distance between the cameras are typically between 2 and 3 m. Fig. 5 illustrates the estimated camera arrangement that was used during a capture. The algorithm properly retrieves the organization of the capture system and the relative distance and orientation of the cameras.

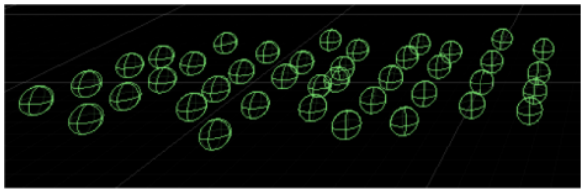

Figure 5: Example of estimated camera position.

4 DESCRIPTION OF THE DATASET

Based on these developed tools, we have built a complete dataset that we share on the following website:

https://project.inria.fr/ftv360.

The dataset is made of several *Captures* including the following steps. (i) We position 40 omnidirectional cameras in a scene. Their distance with the neighboring ones lies between 1m and 3m. (ii) We record one or several calibration sequences, in which a chessboard pattern is moving in the scene. The recorded videos are then used to estimate the calibration parameters with our proposed algorithm. (iii) We record several *Sequences* with the same camera arrangement (and thus the same calibration parameters). In each sequence, a scene (1 min to 4 min) is acquired by all the synchronized cameras. Our dataset is made of three different captures, with, in total 8 different sequences (each of them having 39 or 40 synchronized videos). The structure of the shared dataset is depicted in Fig. 6. We precise that the camera parameters include extrinsic and intrinsic parameters. The extrinsic parameters are given both in the Euler and Rodrigues formats. A complete webpage of the site is dedicated to an explication of what these parameters mean[1]. We also share, for each of the capture, a map of the camera positioning in the scene. Finally, the calibration toolkit is also made available on the project website.

5 CONCLUSION AND FURTHER RESEARCHES

In this paper, we propose a new dataset for supporting researches on FTV. For that purpose, a new acquisition system and a complete calibration solution have been developed. The 8 videos sequences along with the calibration toolkit are made publicly available on *https://project.inria.fr/ftv360*.

These data can serve for the development of new tools for FTV. In particular, virtual view synthesis algorithms are the ultimate step before real 6DoF, where a user could smoothly navigate in the scene. In the same spirit researches on depth estimation, super resolution and inpainting could benefit from this innovative dataset especially because of the original use of multiple omnidirectional cameras. Finally, researchers working on interactive video compression will, for the first time, be able to test their new solutions on a meaningful dataset, *i.e.*, a high number of viewpoints spread in a large scene.

REFERENCES

[1] A. Alatan, Y Yemez, U Güdükbay, X. Zabulis, K Müller, CE Erdem, C. Weigel, and A. Smolic. 2007. Scene Representation Technologies for 3DTV - A Survey. *IEEE Trans. on Circ. and Syst. for Video Technology* 17 (2007), 1587–1605.
[2] Jonathan Courbon, Youcef Mezouar, and Philippe Martinet. 2012. Evaluation of the unified model of the sphere for fisheye cameras in robotic applications. *Advanced Robotics* 26, 8-9 (2012), 947–967.
[3] Ricardo L de Queiroz and Philip A Chou. 2016. Compression of 3d point clouds using a region-adaptive hierarchical transform. *IEEE Transactions on Image Processing* 25, 8 (2016), 3947–3956.
[4] Elsa Dupraz, Thomas Maugey, Aline Roumy, and Michel Kieffer. 2016. Rate-storage regions for Massive Random Access. *arXiv preprint arXiv:1612.07163* (2016).
[5] Olivier Faugeras. 1993. *Three-dimensional computer vision: a geometric viewpoint.* MIT press.

[1] https://project.inria.fr/ftv360/informations/calibration-parameters/

Figure 6: Structure of the FTV360 dataset.

[6] Vincent Jantet. 2012. *Layered Depth Images for Multi-View Coding.* Ph.D. Dissertation. Univ. Rennes 1.

[7] T. Maugey, A. Ortega, and P. Frossard. 2015. Graph-based representation for multiview image geometry. *IEEE Transactions on Image Processing* 24 (2015), 1573–1586.

[8] C. Mei and P. Rives. 2007. Single View Point Omnidirectional Camera Calibration from Planar Grids. In *IEEE International Conference on Robotics and Automation.* Roma, Italy, 3945–3950. https://doi.org/10.1109/ROBOT.2007.364084

[9] A. Roumy and T. Maugey. 2015. Universal lossless coding with random user access: the cost of interactivity. In *Proc. IEEE Int. Conf. on Image Processing.* Quebec, Canada.

[10] U. Takyar, T. Maugey, and P. Frossard. 2014. Extended Layered Depth Image Representation in Multiview Navigation. *accepted in Signal Processing Letters* 21 (Jan. 2014), 22–25.

[11] M. Tanimoto. 2012. FTV: Free-viewpoint Television. *IEEE Signal Processing Magazine* 27, 6 (Jul. 2012), 555–âĂŞ570.